A SPOKEN NATURAL LANGUAGE INTERFACE TO LIBRARIES

Bruce Lund, William M. Fisher, John S. Garofolo, David S. Pallett, Mark Przybocki, R. Allen Wilkinson

National Institute of Standards and Technology (NIST)
Room A216, Building 225 (Technology)
Gaithersburg, MD 20899

ABSTRACT

One way to provide the general public with widespread accessibility to national information networks is by means of spoken natural language interfaces. Spoken natural language is particularly attractive as an interface modality because it is perhaps the most natural way to communicate and it does not require the use of keyboards and display screens. While a wide variety of information will ultimately become available online, one source that is available now and is of great interest to many is found in online catalogs for public libraries. Currently, users of these electronic catalogs must be familiar with various menu-driven access programs involving keyboards and special-purpose commands. There is little doubt that in the near future, people seeking information from libraries will expect to use more sophisticated retrieval software and voice command capabilities that are becoming available on personal computers.

We are currently building a prototype spoken library catalog interface from a technology base of ARPA-sponsored spoken language systems research (e.g. Air Travel Information System [ATIS]) technology [1]. Our project has three purposes: (1) to demonstrate in general the feasibility of spoken natural language interfaces for information services to a new and potentially wide audience, (2) to provide a specific application of this technology to online library catalogs in order to discover the essential technical problems applicable to all such systems (e.g. interfacing to existing library databases), and (3) to promote the development of techniques such as the recognition of names and words which the system has not heard before.

1. OVERVIEW

The NIST Spoken Natural Language Processing Group is developing a speech-based natural language interface to general circulation library card catalogs. In its completed form, the library interface will allow a person to ask a computer system, in plain spoken English, about the presence of books in a library's card catalog. For the project's speech recognition technology, NIST is collaborating with Carnegie Mellon University. The card catalog we use is a subset of the master catalog of the Library of Congress.

Our prototype consists of a speech recognition component, a natural language processing component, and a database retrieval component.

The system uses two approaches to user input. The first is a relatively free-form speech interface which does not require users to specify the class of information they are seeking. A sample request would be, "Show me books on the Civil War published after 1976." This interface parses the input stream of words from the speech recognizer into a canonical frame representing author, title, and subject information as well as possible constraints dealing with publication years.

In the second approach, users tell the system what type of information they are searching for: author, title, or subject. (In this respect it is similar to current, keyboard-based library interfaces.) For instance, users might first say "titles" and then be shown an appropriate template, at which point they can speak a book's title. Again, a canonical frame containing information supplied by the user is produced.

In either approach, the system consults its database and displays, either in a list form or in the form of a simulated catalog card, the information that was requested. Often, the information requested is not literally in the database, and so the use of synonyms and cross-references is being explored. For instance, in our system, all "composers" are "musicians" but the reverse is not true.

The project consists of three phases. In Phase One, completed in November 1994, we demonstrated the ability of the natural language processor and the database to retrieve a subset of records chosen from the Library of Congress' database. In this phase, no spoken language component was used; people typed natural language queries on a keyboard. The purpose of Phase One was to expose the issues involved in any such spoken language interface project that are unrelated to specific database content. Issues which arose in Phase One included:

What sort of questions do users ask?
How do the questions rely on context (e.g. anaphora)?
What sort of output should be presented to users?

We wanted to select a subset of the Library of Congress' database that meaningfully reflected the interests of non-technical users.

We incorporated the recommendations made in the latest *Public Library Catalog* and *Fiction Catalog* (H.W. Wilson Co., publisher), references which suggest books for core library collections. These suggested books helped us to build a database of about 15,000 fiction and nonfiction entries. These entries provide a principled collection which should represent a reasonable balance between accessibility and coverage.

In Phase Two, currently underway, we are incorporating speech recognition. Essentially, Carnegie Mellon University's Sphinx-II speech recognizer is pipelined, via a client/server connection, to our natural language processing and database retrieval components. The speech recognizer's vocabulary is the same one used by Phase One's natural language component.

Phase Three will involve testing in an actual library environment. We will install the computer equipment in an information kiosk in a local library. A database of books particular to that library can be extracted from the Library of Congress catalog. The language model for the speech recognizer can then be generated from the database. The kiosk system will record the questions of the users, which can be used to further refine the system's language model and natural language parser.

Ultimately, we expect this project to permit an interface to not only traditional textual but also to networked digital library databases. Our interface will be increasingly reliant on spoken language for both input and output.

1.1 Natural Language Component Overview

The natural language component which processes user input, whether from a keyboard or the Sphinx recognizer, consists of a parser and lexicon.

Parser. To convert the user's natural language query into a canonical form by which the search module can explore the database, one of several parsers can be used. We have been experimenting with both the Carnegie Mellon Phoenix transition network parser and a definite clause grammar parser of our own design. Regardless of what parser is used, the output is a canonical database query taking the form:

```
request_type(request_action,
  constraint(type(value),work),
  author/title/subject)
```

The request type can be "author," "title," "subject," or the compound requests "author-title" and "author-subject." The request action can be "find" (to display books) or "count" (used, for example, when asked how many books a certain author wrote). Valid constraint types are "before" or "after" a given year, "earliest" or "latest", and "all" (no constraint). The work is always "book," although the Library of Congress has other types of works in its master database. The last element of a query is the string of author, title, and subject information supplied by the user.

The parser retains a record of all queries so that context can be provided if necessary. For example, assume the user types, "Show me all books by Mark Twain." The parser will convert this question into the following canonical database query:

```
author_request(find,
   constraint(all,work(book)),
   author(query_auth(twain,mark))).
```

If the user then types "Only the latest," the parser will discover that the only meaningful information in the query is "only" and "latest" and then apply these constraints to the most recent query. A new database query will be produced reflecting the new constraints:

```
author_request(find,
   constraint(latest,work(book)),
   author(query_auth(twain,mark))).
```

The system currently under development correctly answers requests such as:

1. "Show me Tom Sawyer by Mark Twain published after 1988."
2. "List books dealing with science fiction by Clarke."
3. "I'd like to see the last book by Isaac Asimov"
4. "Do you have, aah, do you have books on trees?"

Robustness is a crucial feature of any system designed for public use. Special attention is being paid to allow for restarts (seen in sentence 4) and out of vocabulary words. If a query is partially understood, but not well enough to provide a database query, the system responds with "Sorry, could not understand [out of vocabulary word(s)]". If a query is understood but the information is not in the database (e.g. books by Faulkner published in 1995), the system responds with "This book is not in the database."

Lexicon. The database of Library of Congress records (which are in the USMARC format) is processed by software which extracts author names, titles, and subject information into three respective lexicons. In addition, the words that people use to form their queries (e.g. show, what, only) are incorporated.

The current version of the NL component uses lexicon files with the following information (approximate counts are given):

12,700 unique titles (consisting of 19,000 tokens)
6,000 unique author names (5,800 tokens)
6,000 unique subjects (4,200 tokens)

These were extracted from a prototype database of approximately 14,000 records.

1.2 Database Component Overview

The system has a database component consisting of a search module and database. When records are retrieved, users are presented with bibliographic information by means of a display module. Each of these is described below.

Search Module. The search module is the interface between the parser and the database. Its purpose is to convert the parser's output into a form that will result in the most informative and efficient response from the database. Since a database of even a few thousand books occupies several megabytes, indexes are used to enhance searches for the data that a user is most likely to request: author names, titles, and subjects.

The parser presents database queries to the search module of such request types (e.g. author, title, subject, author-title, author-subject). The latter two are produced when a user asks questions like "Show me Huckleberry Finn by Mark Twain" or "Show me books by Twain on steamboats." Constraints such as "latest" or "before 1965" also become part of the canonical query.

While these types of searches are relatively straightforward because they involve simple pattern matching techniques, more challenging cases arise when inexact matching is required. There are two varieties: fragment matching and conceptual matching.

In the first, we need a way to deal with string fragments if, say, an author's first name is either mispronounced or misrecognized (or simply incorrectly given by the user). For example, in a recent test the speech recognizer presented the NL component with the string "Books by Pearson Keillor" (the user actually said "Garrison Keillor"). If a complete author's name does not match what is in the database, but the last name does, then all books by that author will be retrieved. A similar technique is used to match titles and subjects. In this case, all books by authors named Keillor were retrieved.

The second required variety of inexact matching is a conceptual match. For example, if the user asks, "Display books about international relations," some relevant books will not be displayed if exact or approximate string matching techniques are used, because the string "international relations" does not necessarily occur in the Subjects field of a relevant book's Library of Congress database entry. In these cases, no amount of fragment matching can suffice. Instead, a conceptual hierarchy is needed.

The Dewey Decimal Classification system provides a straightforward hierarchy which can be quite useful to the library interface. In the above example, books on international relations would indeed have been found by a search module which has access to the Dewey hierarchy (under which the desired category has Dewey Code 327). The natural language module currently has a database of Dewey codes at the "tens level" (e.g. 300 = Social Sciences, 310 = Statistics, 320 = Political Science, etc.). More fine-grained coding at the "ones level" (e.g. 326 = Slavery & emancipation, 327 = International relations, 328 = Legislation) can be employed if necessary.

However, there are limitations to the Dewey system approach. For example, it lacks the conceptual richness of a semantic network such as WordNet [2]. Also, its 19th century origin skews it toward quite a different world. For these reasons, we also make use of the Library of Congress classification schedules which are similar to the Dewey system, yet more comprehensive, especially in scientific and technical subdivisions. Since neither classification system can completely replace the other, both are used. Since a given book can be identified in a subject search using both the Dewey and the LC classifications, redundant "hits" are removed before they are displayed to the user.

Database. The bibliographic information we obtained from the Library of Congress consisted of about 1.5 gigabytes of data in the USMARC format, representing about two million titles. We had to perform several operations to arrive at the current set of approximately 14,000 titles recommended in the Wilson core collection reference books. We also needed their associated data fields (e.g. author, subject, and publication information, classification numbers, etc.). Unfortunately, no on-line version of these reference books was available to us, so we chose to cut the books apart, scan them, and then use an optical character recognition program to obtain a machine-readable listing of titles, authors, ISBN codes, and Library of Congress classification numbers for the desired books.

However, the OCR process itself is prone to inaccuracies due to various physical factors such as print and paper. In addition, not every entry had all of the above-mentioned fields. When any two fields matched an entry in the master Library of Congress database, it was assumed to be a valid entry and so the full Library of Congress record was retrieved. The most efficient retrieval order turned out to be Library of Congress classification number (identified as USMARC tag 050), ISBN code (tag 020), author (tag 100), and finally title (tag 245).

It was also found that there were too many editions of numerous books, so the database was manually pruned. We used our judgement in deciding which books should have more than one edition listed, but our general rule was to keep hardcovers and the most recent editions.

Eventually, the data extraction process produced a database which had about 14,000 complete Library of Congress card catalog listings. Here is a sample from our master database for the book *The Big Sleep*:

```
book(1633,001,'76011809 //r932').
book(1633,003,'DLC').
book(1633,005,'19930319172054.3').
book(1633,010,'76011809 //r932').
book(1633,020,'0394721365').
book(1633,040,'DLC').
book(1633,040,'DLC').
book(1633,050,'PZ3.C3639').
book(1633,050,'Bi9').
book(1633,050,'PS3505.H3224').
book(1633,082,'813/.5/2').
book(1633,100,'Chandler, Raymond,').
book(1633,100,'1888-1959.').
book(1633,245,'The big sleep /').
```

```
book(1633,245,'Raymond Chandler.').
book(1633,260,'New York :').
book(1633,260,'Vintage Books,').
book(1633,260,'1976, c1939.').
book(1633,300,'216 p. ;').
book(1633,300,'18 cm.').
book(1633,650,'Marlowe, Philip (Fictitious
character)').
book(1633,650,'Fiction.').
book(1633,650,'Private investigators').
book(1633,650,'California').
book(1633,650,'Los Angeles').
book(1633,650,'Fiction.').
book(1633,655,'Detective and mystery sto-
ries.').
book(1633,655,'gsafd').
```

The first field of each line (having the value 1633 in this example) is the book's unique identification number in our system. The second field (e.g. 100) is the USMARC tag number for that field (100 identifies authors). The third field is text assigned by the Library of Congress. The database currently is in Prolog predicate form, although we structured it to allow simple conversion to SQL-accessible tables.

From this master database, we extracted author, title, and subject fields which were then used by Carnegie Mellon University to build a language model for the Sphinx-II recognizer. The natural language parsing component also uses these three fields to build the canonical database query representing a user's request.

Display Module. The final component of the library interface system presents output to the user by means of simulated library catalog cards displayed on an X Windows computer screen. Four such cards will fit on a 19" monitor, with a supplementary window below in which the system displays such non-catalog information as error messages. Additionally, the user can switch to a listing mode in which 28 book entries can appear on one screen. Scroll buttons allow back and forth navigation if there are too many books to display on one screen.

2. CHALLENGES

The developers of spoken natural language interface systems face special problems. For example, the perplexity is high in classes such as author names or titles. Performance will be enhanced if clues from the context of a request can be used to restrict the amount of searching required. One way to give the system clues that, say, the user is asking about authors and not titles, would be to use a more restricted interface in which users choose the class of search they want to make before giving the request, or to implement other dialogue-management and mixed-initiative strategies.

Library catalogs must be updated periodically. Thus another problem in developing such a system is dealing with names, subjects, and titles that the system has not heard before. Probably the most efficient way to deal with this problem is to automatically generate pronunciations from existing dictionaries, and then to incorporate them into the system's models.

One way of dealing with unknown names and words while the system is in operation is to provide a "spell mode." This would allow users to pronounce the individual letters of names and words that the system is having trouble identifying.

Finally, the system must be able to elicit additional information from the user by means of on-screen dialogues when it needs to clarify a request. For example, if a user asks for books on the Civil War, the system can ask if only the fiction books should be displayed (e.g. *Gone with the Wind*), just the nonfiction ones, or all.

3. FUTURE DIRECTIONS

NIST hopes to work with others in starting work on a Phase Three prototype system in CY 1995. It may consist of a library catalog information kiosk with a telephone handset (rather than a headset-mounted microphone) for user input along with a suitable video display. A possible addition would be a speech synthesizer to complement the display screen. Instead of simply reading the same information that is available on the cards, it could provide system-level information such as the number of books that match the query, as well as prompts to the user.

More significantly, the speech output will provide the framework for testing our user interface software which will gradually rely less on the display screen and more on spoken words for output, paving the way for the project's ultimate goal, a completely spoken natural language library interface.

During the development, NIST plans to use the prototype to collect a corpus of data and make it, along with our library catalog database, available to the spoken language research community. We envision developers eventually transferring this technology to enlarged domains in which their systems would work with the existing and future information service database applications.

4. ACKNOWLEDGEMENTS

We wish to acknowledge the assistance of Sunil Issar of Carnegie Mellon University who ported the Sphinx-II speech recognizer to the library catalog domain. He also adapted the Phoenix parser, which was used in an early version of our project, to this domain.

REFERENCES

[1] Dahl, D., et al. (1994) "Expanding the Scope of the ATIS Task: The ATIS-3 Corpus", in *Proceedings of the Human Language Technology Workshop*, March 1994 (Weinstein, C.J., ed.).

[2] Miller, G. A., Ed. (1990) "Five Papers on WordNet." *International Journal of Lexicology*, 3, No. 4 (Revised March 1993).